



Positioning with DHTML

Objectives

- ▶ Understand DHTML positioning
- ▶ Position an element absolutely
- ▶ Position an element relatively
- ▶ Size an element manually
- ▶ Stack screen elements
- ▶ Add a scroll bar
- ▶ Create a sidebar
- ▶ Incorporate an advanced positioning function

One of DHTML's greatest contributions to Web page design is a tool for **positioning**, or specifying the precise placement of elements within the page. Just like other DHTML components, DHTML positioning opens doors to many possibilities for new Web page features. Other DHTML features, such as scroll bars, complement positioning to help create effective page layouts.  Lydia wants to enrich the Web page design for the Nomad Ltd Web publication. She will use positioning and other DHTML layout features to create a sophisticated, attractive style.





Understanding DHTML Positioning

A fundamental difference between document layouts in traditional media, such as posters and magazines, and document layouts in HTML is HTML's lack of tools for precise placement of page elements. DHTML allows precise positioning of page elements through an extension of cascading style sheets called **Cascading Style Sheets - Positioning (CSS-P)**. CSS-P allows you to position elements either **absolutely**, at fixed coordinates on a user's screen, or **relatively**, based on the position of other screen elements. To specify positioning, you use the **position** attribute, which is a style sheet property. Although some advanced page layout is possible with basic HTML, CSS-P makes the task much easier to code and offers features not possible with HTML alone. As Lydia researches CSS-P, she learns about several new features that she would like to include in her Web pages, including columns, overlap, and scripted effects.

Details



Columns

Many Web page designers have created advanced layout features using basic HTML formatting. For example, HTML-only pages can use tables to display text in columns, rather than in one single block. However, adding these features in HTML can be difficult and limiting because the tags were not designed originally to provide advanced formatting. CSS-P makes this type of formatting much simpler by allowing you to easily specify each element's width and location on a Web page. CSS-P also places elements more predictably in different screen resolutions. Lydia plans to use the CSS-P float feature to add a sidebar to the Nomad Ltd tents page, similar to the one shown in Figure L-1.



Overlap

A design feature not found in HTML but available in CSS-P, is the ability to overlap screen elements, which facilitates adding labels over graphics. Also, it allows you to create complex layouts such as ones that superimpose words in different colors or that overlap parts of images. The Web page in Figure L-2 uses CSS-P to overlap text and graphics. Lydia wants to use the overlap feature to create a distinctive design effect in her Web pages for Nomad Ltd. She plans to place the general category name for each Web page in large, light-colored text behind the page headings.



Scripted features

As with other DHTML tools, combining CSS-P with scripts allows you to create many new display features for your Web pages. For example, by changing a graphic's dimensions slightly at regular intervals, you can animate with DHTML. You also can use scripting to allow users to drag elements to new positions on the Web page. Lydia plans to add some draggable elements to her tents page to help users visualize the placement of sleeping bags in various tent designs.

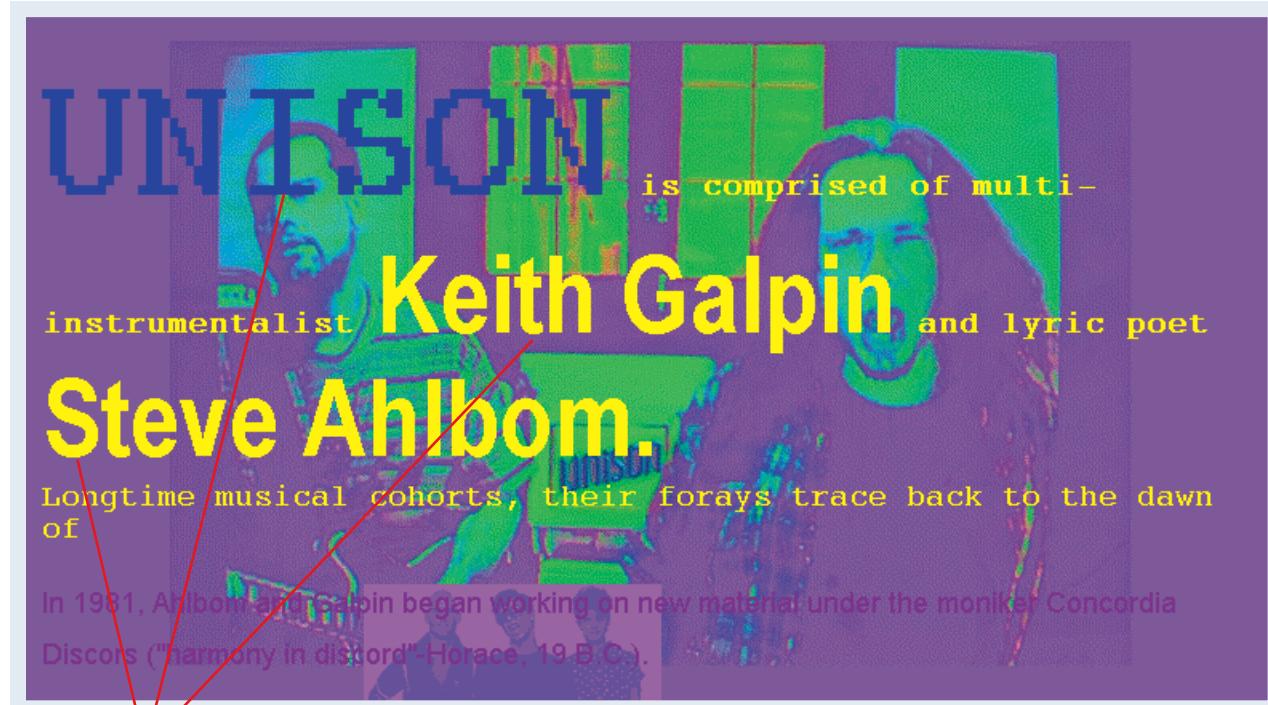
FIGURE L-1: Web page containing sidebar

webmonkey/dynamic_html/

Resources <hr/> Inside DHTML This site by Scott Isaacs, a member of the Internet Explorer design team, provides lots of juicy coverage of Dynamic HTML and excerpts his book - you guessed it, Inside Dynamic HTML . Toolbox <hr/> Dreamweaver <i>Platform:</i> Win 95, Power Mac Cost: not yet set Company: Macromedia	Taylor's Tutorial <hr/> Taylor's Dynamic HTML Tutorial - Day 1 Dynamic HTML is how Netscape's and Microsoft's 4.0 browsers are pushing the Web to new limits. In the first of five parts, Taylor looks at what dHTML is all about and what skills you need to code for it. <i>9 Mar 1998</i> Taylor's Dynamic HTML Tutorial - Day 2 Taylor digs into dynamic HTML, showing you the basics of using CSS-P to lay out your pages. He even looks at the elusive z-index. <i>10 Mar 1998</i> Taylor's Dynamic HTML Tutorial - Day 3 Today Taylor's series gets tricky: By the end of the day, he'll have you scripting dHTML and making monkeys run around
--	--

Sidebar created using CSS-P

FIGURE L-2: Web page displaying element overlap





Positioning an Element Absolutely

With CSS-P, you can specify an element's position in several different ways. The most straightforward way is to use **absolute positioning**, which lets you specify the left and top coordinates of an element on the Web page. You use the CSS-P **left** and **top** properties to specify an element's location relative to the top-left corner of its **parent**, or the object enclosing it. For example, a element nested within <DIV> tags would be positioned relative to the <DIV> element, its parent. In this case, the element is known as a **child** of the <DIV> element. Any element not enclosed by another element is a child of the browser window and is absolutely positioned with respect to the top-left corner of the window. You can specify left and top values in points (pt), pixels (px), inches (in), millimeters (mm), or centimeters (cm). If you don't specify units, the browser defaults to **pixels**, which are the tiny units of light that create the display on a monitor. The number of pixels visible on a user's screen varies depending on its resolution. However, even when using pixels, it is a good idea to specify units in order to make your code clearer when debugging it and when others read it.  As she develops a Web page describing tents available from Nomad Ltd, Lydia wants to reposition the elements located at the top of the page to decrease the amount of blank space in her original design. She adds absolute position information to the elements to create a more compact layout.

Steps 123

1. Start your Web browser, cancel any dial-up activities, then open the file **HTML L-1.htm**

Notice how the text flows around the image. In this version of the Web page, because Lydia was using only HTML, she made the text flow by grouping the image and the text together in a DIV tagset. However, text flow created in HTML can be unpredictable depending on the user's display font size and resolution settings.

2. Start your text editor program, open the file **HTML L-1.htm**, then save it as a text document with the filename **Tent absolute position.htm**
3. Scroll to the document's HEAD section, select the text **[replace with logo absolute position code]** in the embedded style sheet, then press **[Delete]**
4. Type **#logo {position: absolute; top: 30px; left: 30px}**

The name #logo associates this style with the ID "logo". Figure L-3 shows the document containing the absolute position code. Lydia placed the logo so that it is 30 pixels from the top of the browser window and 30 pixels from the left edge of the browser window. By positioning the Nomad Ltd logo using absolute positioning, Lydia's headers can move up to fill in the empty area at the top of the page.

5. Scroll down until the opening DIV tag above the IMG tag for nomad.jpg appears in the document window, select the text **[replace with ID]**, then press **[Delete]**
Lydia wants her layout to work on both fourth-generation browsers. Because Netscape Navigator does not recognize absolute positioning referenced in an IMG tag, she has enclosed the IMG tag in DIV tags. By referencing the position information in the DIV element, Lydia positions the graphic element.
6. Type **ID="logo"**
7. Check your document for errors, make any necessary changes, then save **Tent absolute position.htm** as a text document

8. Open **Tent absolute position.htm** in your Web browser

As Figure L-4 shows, the Nomad Ltd logo appears in the top-left corner of the window, moved slightly down and to the right from its position in the file HTML L-1.htm. Because the graphic is placed absolutely at a fixed location in the window, the remaining screen elements flow beginning at the top of the window. The presence of the graphic to the left of the headings has no effect on their alignment or flow. As Figure L-4 shows, the result is overlap between the logo and the heading text. Next, Lydia will adjust the position of the heading text to keep it from overlapping the logo.

QuickTip

Determining the exact number of units for an element's position coordinates is a process of trial and error.

Trouble?

The text flow may appear differently on your screen, depending on its resolution.

FIGURE L-3: Web document containing absolute position code

```
<HTML>
<HEAD>

<TITLE>Nomad Ltd - Selecting a tent</TITLE>

<LINK REL=stylesheet HREF="nomadltd.css" TYPE="text/css">

<STYLE>
<!--
.tenthead {font-family: arial, sans-serif; font-size: 14pt}
.button {font-family: impact, arial; font-size: 8pt}
.norm {font-weight: normal}
.noital {font-style: normal}
#logo {position: absolute; top: 30px; left: 30px}
//-->
</STYLE>

<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

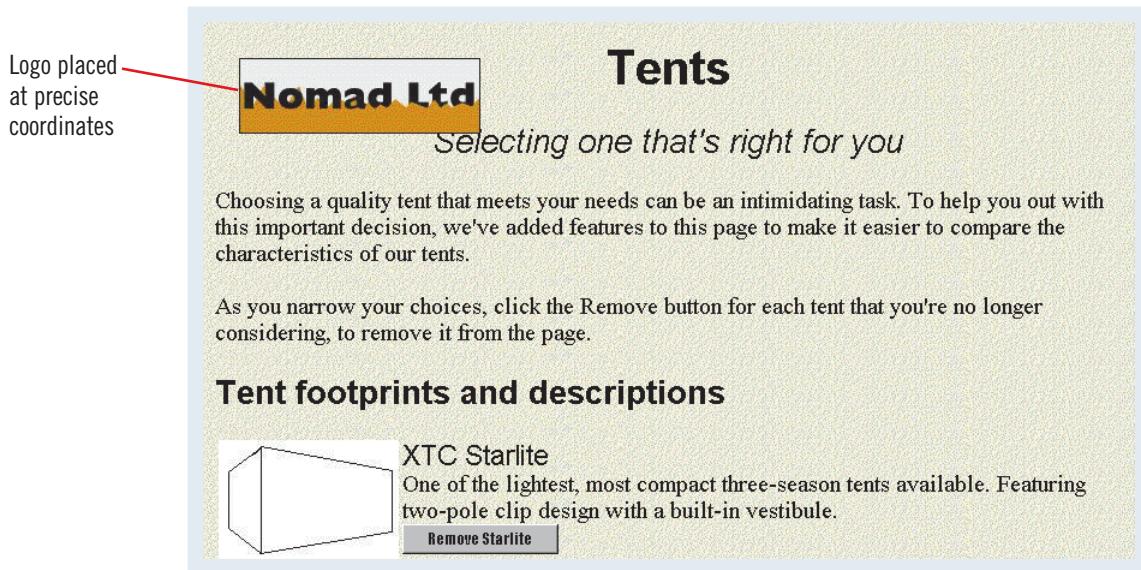
if(!IE4) {window.onerror=null}

totalTents = 0;
function countHeaders() {

Name associates style with ID "logo"
```

Code to position logo at top-left corner of page

FIGURE L-4: Web page displaying absolutely positioned logo



Cross-browser positioning

Both fourth-generation browsers interpret and display CSS-P formatting, but as with the other parts of DHTML, each browser processes CSS-P differently. This disparity results in unique code to create features in each browser that you need to remember when creating cross-browser code. The main challenge is that Navigator 4 does not correctly interpret positioning information inserted directly in a tag. In

fact, inline coding for position removes the style information from all elements in the page that follow the inline code. You can remedy this problem by defining all your positioning code in either embedded or external style sheets. Fortunately, grouping style information at the top or in a separate file brings other benefits because it organizes your code and makes it easier to read and understand.



Positioning an Element Relatively

In addition to placing elements at fixed screen coordinates, CSS-P allows you to simply offset elements from their default positions in the page flow. This format, called **relative positioning**, is useful when you want your document to always display an element before or after other elements, but at a specified horizontal or vertical offset. Lydia wants to indent the page headings, while leaving them in the general page flow. She uses relative positioning to specify the new placement for the headings.

Steps 123⁴

1. Open the file **HTML L-2.htm** in your text editor, then save it as a text document with the filename **Tent relative position.htm**
2. In the embedded style sheet, select the text **[replace with head relative position code]**, then press **[Delete]**
3. Type **#head {position: relative; left: 250px}**

Figure L-5 shows the document containing the relative position code. Similar to the absolute position of the logo, the left property that Lydia used to position the text moves it left in relation to the parent element, which is the browser window. Absolute positioning removes an element from the flow of the document, which causes the elements below it to move up and to overlap its former position in the page flow. This caused the headings to move up in the last lesson. Relatively positioning leaves an element in the document flow. A relatively positioned element moves relative to its default location in the page, but the elements that follow a relatively positioned element do not move up to take that position. This is what Lydia wants to do in her document because she wants the headings indented from the left to appear next to the graphic, but she does not want the text that follows to overlap the headings and graphic.

4. Scroll until **<DIV ALIGN="center"** appears in the document window, select the text **[replace with ID]**, then press **[Delete]**
5. Select the adjacent text **ALIGN="center"** and press **[Delete]**
6. Type **ID="head"**

Lydia references the position information in the DIV element to position the headings.

7. Check your document for errors, make any necessary changes, then save **Tent relative position.htm** as a text document
8. Open **Tent relative position.htm** in your Web browser

As Figure L-6 shows, the main heading and the subheading are indented far enough from the left edge of the window to allow room for the logo. Because the indent was specified using relative positioning, the text after the headings does not move up into the positions previously held by the headings but, rather, continues to flow below them.

FIGURE L-5: Relative position code in Web document

Code to
position
heading
within
document
flow

```
<HTML>
<HEAD>

<TITLE>Nomad Ltd - Selecting a tent</TITLE>

<LINK REL=stylesheet HREF="nomadltd.css" TYPE="text/css">

<STYLE>
<!--
.tenthead {font-family: arial, sans-serif; font-size: 14pt}
button {font-family: impact, arial; font-size: 8pt}
.norm {font-weight: normal}
.noital {font-style: normal}
#logo {position: absolute; top: 30px; left: 30px}
#head {position: relative; left: 250px}
//-->
</STYLE>

<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

if(!IE4) {window.onerror=null}

totalTents = 0;
```

FIGURE L-6: Web page displaying relatively positioned headings

Headings
indented
within
main text
flow

The screenshot shows a web page with the following structure:

- Header:** A yellow box containing the text "Nomad Ltd".
- Title:** "Tents" (bold)
- Text:** "Selecting one that's right for you"
- Text:** "Choosing a quality tent that meets your needs can be an intimidating task. To help you out with this important decision, we've added features to this page to make it easier to compare the characteristics of our tents."
- Text:** "As you narrow your choices, click the Remove button for each tent that you're no longer considering, to remove it from the page."
- Section:** "Tent footprints and descriptions"
- Image:** A small 3D wireframe drawing of a tent.
- Text:** "XTC Starlite"
- Text:** "One of the lightest, most compact three-season tents available. Featuring two-pole clip design with a built-in vestibule."
- Button:** "Remove Starlite"



Sizing an Element Manually

In addition to position on the page, DHTML style properties allow you to specify an element's dimensions using the **height** and **width** properties. You can specify the two dimensions separately by using the same units available for the positioning properties. Additionally, you can size the element relative to its parent by using percentages. If you choose not to specify the height or the width, the browser sizes the element automatically.  Lydia wants to reformat the description text for each tent model, so it displays indented and in a narrower column. Because she is changing style information for several screen elements, she adds properties to the page's embedded style sheet.

Steps 123⁴

1. Open the file **HTML L-3.htm** in your text editor, then save it as a text document with the filename **Tent element size.htm**
2. Select the text **[replace with tentbody class description]** in the embedded style sheet, then press **[Delete]**
3. Type **.tentbody {position: relative; left: 100px; width: 300px}**
Figure L-7 shows the document code including the new width property.
4. Scroll down until the **<DIV ID="tent1" name="tent">** tag appears in the document window, then read through the code within this DIV element
Notice that, in addition to the new style properties, Lydia has inserted **CLASS="tentbody"** in the DIV tag already, which applies the position information from the **tenthead** class description she created. She has added this information for each tent.
5. Check your document for errors, make any necessary changes, then save **Tent element size.htm** as a text document
6. Open **Tent element size.htm** in your Web browser, then scroll down until the tent descriptions appear in the document window
As Figure L-8 shows, each description is indented from the left margin and the paragraph width is narrowed, which creates a column.

FIGURE L-7: Web document including width property code

```
<HTML>
<HEAD>

<TITLE>Nomad Ltd - Selecting a tent</TITLE>

<LINK REL=stylesheet HREF="nomadltd.css" TYPE="text/css">

<STYLE>
<!--
.tenthead {font-family: arial, sans-serif; font-size: 14pt; position: relative; left: 100px}
.button {font-family: impact, arial; font-size: 8pt}
.norm {font-weight: normal}
.noital {font-style: normal}
#logo {position: absolute; top: 30px; left: 30px}
#head {position: relative; left: 250px}
.tentbody {position: relative; left: 100px; width: 300px}
//-->
</STYLE>

<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

if(!IE4) (window.onerror=null)

```

Width property for tentbody class

FIGURE L-8: Web page displaying adjusted width

As you narrow your choices, click the Remove button for each tent that you're no longer considering, to remove it from the page.

Tent footprints and descriptions

List indented

Column narrowed with width property

XTC Starlite
One of the lightest, most compact three-season tents available.
Featuring two-pole clip design with a built-in vestibule.

Amano Brevifolia
The simple, vaulted design characterized by two doors and two vestibules returns with the 2000 Brevifolia model. New features



Positioning and sizing using percentages

Although standard measurement units, such as pixels and points, are most familiar to Web page designers, the ability to use percentage as a positioning and sizing unit offers advantages for some screen elements. Because monitors of different screen resolutions display the document window with a larger or smaller area, a size in pixels or points appears at a different position on the screen at different resolutions. While a column of 150px may fit perfectly in an 800 x 600 display, that same column may be surrounded by space at 1024 x 768, reducing the effectiveness of your

layout. Elements sized with percentages, however, automatically adjust to the size of their parent elements. Thus, if you size a graphic element that is a child of the document window at 35%, that graphic element will maintain the same size relative to the document window in different resolutions. You can use the same method to absolutely or relatively position an element as a percentage of its parent. For some applications, specifying an exact measurement is important, but percentage sizing and positioning are important tools in your Web page design toolbox.



Stacking Screen Elements

Because an absolutely positioned element can appear anywhere on a Web page, including the space occupied by other elements, the browser can't format pages containing these elements as it would standard HTML pages. Instead, each absolutely positioned element is considered to be on a separate **layer**, which is a transparent virtual page that determines overlap order. Web page layers are like sheets of clear plastic with writing or images on them in different areas. When the sheets are superimposed, as layers are in the browser window, all the contents of all sheets are visible; but some contents may block out others, depending on their order in the stack. Each layer's **z-index** property determines its position in the stack. Higher numbers are located closer to the top of the stack, and elements on these layers will block out elements in the same position on lower layers of the stack. An element that is positioned using absolute positioning is placed on a separate layer. An element positioned using relative positioning remains on the same layer as the rest of the standard page elements. Lydia plans to label each page in Nomad Ltd's Web publication based on its content category. She wants to place the category name at the top of the page so that it appears behind the headings. For the tent page, she wants to add the word *camping* to the heading background in large, light-colored type.

Steps 123

1. Open the file **HTML L-4.htm** in your text editor, then save it as a text document with the filename **Tent layers.htm**
2. In the embedded style sheet, select the text **[replace with backtext layer code]**, then press **[Delete]**
3. Type **#backtext {position: absolute; left: 250px; font-size: 64pt; font-family: arial; color: #7093DB; z-index: -1}**
4. Scroll down and select the text **[replace with background text]**, then press **[Delete]**
5. Type **<DIV ID="backtext">** and press **[Enter]**
6. Type **CAMPING** and press **[Enter]**, then type **</DIV>**
7. Check your document for errors, make any necessary changes, then save **Tent layers.htm** as a text document
8. Open **Tent layers.htm** in your Web browser

As Figure L-10 shows, the text *CAMPING* appears behind the headings in a large and light-colored font. This stacked layout allows Lydia to add extra information to the Web page without disrupting the flow of the page. It also adds an interesting, unusual visual effect.

FIGURE L-9: Web document including background text code

```
<HTML>
<HEAD>

<TITLE>Nomad Ltd - Selecting a tent</TITLE>

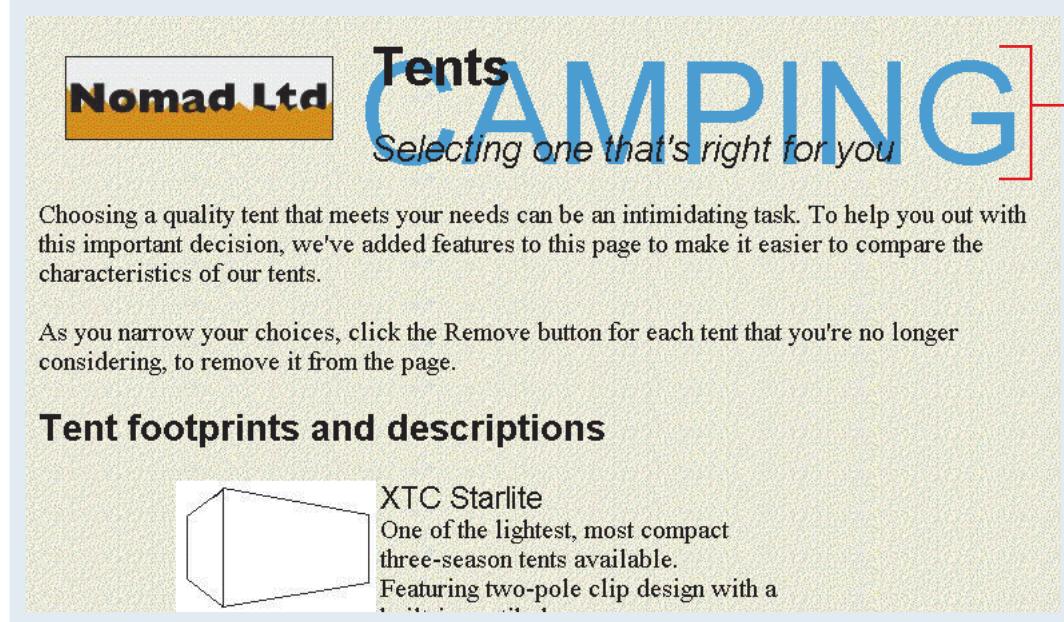
<LINK REL=stylesheet HREF="nomadltd.css" TYPE="text/css">

<STYLE>
<!--
.tenthead {font-family: arial, sans-serif; font-size: 14pt; position: relative; left: 100px}
.button {font-family: impact, arial; font-size: 8pt}
.norm {font-weight: normal}
.noital {font-style: normal}
#logo {position: absolute; top: 30px; left: 30px}
#head {position: relative; left: 250px}
.tentbody {position: relative; left: 100px; width: 300px}
#backtext {position: absolute; left: 250px; font-size: 64pt; font-family: arial; color: #7093DB; z-index: -1}
//-->
</STYLE>

<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

z-index property for background text
```

FIGURE L-10: Web page displaying background text





Adding a Scroll Bar

You can use CSS-P to associate a scroll bar with an element when the element is too large to fit its defined size. This effect, which you create using the **overflow** property, allows you to create the equivalent of an independent frame anywhere within your browser window. Of the fourth-generation browsers, only Internet Explorer accommodates the overflow property.  To make the tent page layout more concise, Lydia formats the list of tent outlines and descriptions in a box with a scroll bar. This allows users to scroll from top to bottom in the page more quickly and still easily view the tent descriptions if they wish.

Steps 123⁴

1. Open the file **HTML L-5.htm** in your text editor, then save it as a text document with the filename **Tent scroll.htm**
2. In the embedded style sheet, select the text **[replace with list scroll code]**, then press **[Delete]**
3. Type **#list {height: 300px; width: 600px; overflow: auto}**

Figure L-11 shows the Web page code in the embedded style sheet. Because the list of tent descriptions is much longer than 300 pixels, the height specification creates a display area smaller than the object size. The “auto” value for the “overflow” property instructs the browser to display scroll bars only where necessary. Because the text in this case is longer than 300 pixels, the browser will create a vertical scroll bar for the object. The width of this DIV is not constrained by a style setting, so a horizontal scroll bar is not needed.

4. Scroll until the code for the heading Tent footprints and descriptions appears in the document window, select the text **[replace with opening DIV tag]**, then press **[Delete]**
5. Type **<DIV ID="list">**
6. Scroll to the end of the tent description list which is after the description for tent 7, select the text **[replace with closing DIV tag]**, press **[Delete]**, then type **</DIV>**
7. Check your document for errors, make any necessary changes, then save **Tent scroll.htm** as a text document
8. Open **Tent scroll.htm** in your Web browser, then scroll down to view the list of tent outlines and descriptions
As Figure L-12 shows, in IE 4 the tent-description list displays in a limited area with a vertical scroll bar on the right edge.
9. Use the scroll bar for the list to view all of the tent descriptions

Trouble?

The amount of text displayed in your scroll bar box will depend on the size of the font you are using and your screen resolution.

FIGURE L-11: Web document including scroll bar code

```
<HTML>
<HEAD>

<TITLE>Nomad Ltd - Selecting a tent</TITLE>

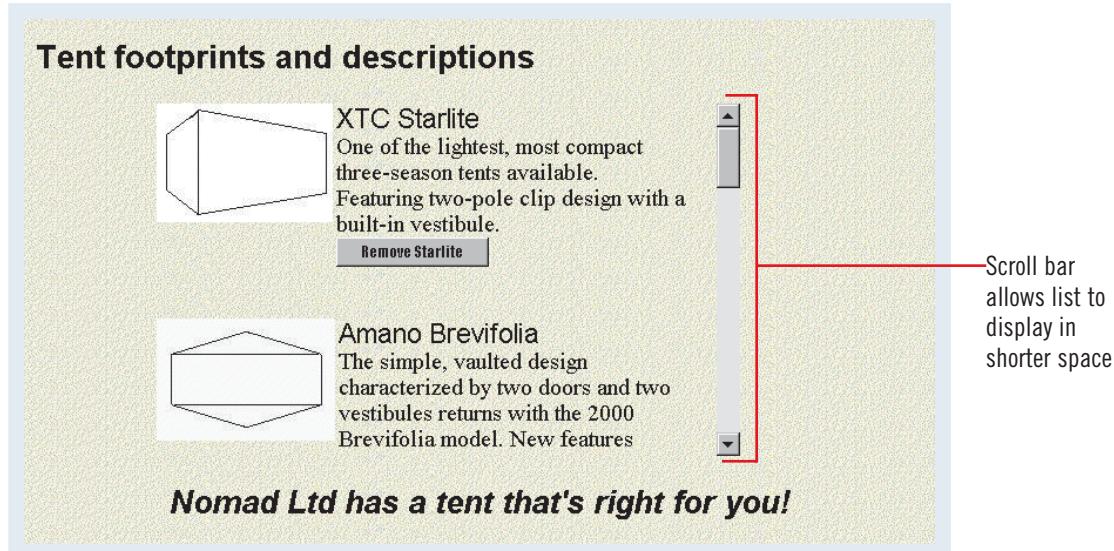
<LINK REL=stylesheet HREF="nomadltd.css" TYPE="text/css">

<STYLE>
<!--
.tenthead {font-family: arial, sans-serif; font-size: 14pt; position: relative; left: 100px}
.button {font-family: impact, arial; font-size: 8pt}
.norm {font-weight: normal}
.noital {font-style: normal}
#logo {position: absolute; top: 30px; left: 30px}
#head {position: relative; left: 250px}
.tentbody {position: relative; left: 100px; width: 300px}
#backtext {position: absolute; left: 250px; font-size: 64pt; font-family: arial; color: #7093DB; z-index: -1}
#list {height: 300px; width: 600px; overflow: auto}
//-->
</STYLE>

<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
```

overflow property adds scroll bar

FIGURE L-12: List formatted with scroll bar



Creating a clip region

Sometimes, you want to put a large element on your Web page but don't have room for the full element; other times, you only want to display a portion of a large element. CSS-P's **clip** property allows you to control how much of an element is visible on your Web page by acting as a layer above the element, covering all of it except for a hole you define, called a **clip region**. (Note that Netscape Navigator 4 does not support this property.) To create a clip region, you specify the coordinates of a rectangle, usually with an

area smaller than the element, using the syntax **clip: rect(top right bottom left)**. The abbreviation "rect" stands for rectangle, which is the only shape currently supported by this property. Substitute each of the terms in parentheses with a coordinate value, or enter **auto** to leave the default. When the element appears in your Web page, the only portion of the element that will be visible is the section within the rectangle you specified.



Creating a Sidebar

Using CSS-P's placement and sizing properties, you can create and position text blocks independently of each other. You can use the **float** property to remove an element from the main text flow and display it to the side of the flow. The **left** and **right** values allow you to specify whether the element is positioned on the right or left side of the main document flow. The float feature allows you to create many text effects, including sidebars, which are difficult to create with HTML alone.  Lydia wants to add scale outlines of backpacks and sleeping bags to the tents page to give users a better feel for the relative sizes of the tents. To make this area stand out from the page's main text, she creates a sidebar.

Steps 123⁴

1. Open the file **HTML L-6.htm** in your text editor, then save it as a text document with the filename **Tent sidebar.htm**
2. In the embedded style sheet, select the text **[replace with sidebar code]**, then press **[Delete]**
3. Type **#sidebar {width: 350px; float: right; position: absolute; left: 400px; font-family: arial; font-size: 11pt; background: #8FBC8F}** and press **[Enter]**

The “float” property removes the section from the document flow, and the “right” value specifies that it floats to the right of the flow. Lydia could specify the height, but the height property works in conjunction with the float property only in Internet Explorer. Navigator always adjusts the height of a sidebar to fit its contents, regardless of the height setting. By not assigning the height property, Lydia allows both browsers to automatically adjust the height to ensure uniform appearance.
4. Type **.expl {width: 375px}**

To keep the paragraph to the left of the sidebar from overlapping it, Lydia associates it with a fixed width. Figure L-13 shows the Web page code containing the new style specifications.
5. Scroll until the DIV tag before the IMG tags appears in the document window, select the text **[replace with ALIGN and ID codes]**, then press **[Delete]**
6. Type **ALIGN="left" ID="sidebar"**

Sidebars that float to the right of the main text also automatically align text along the right edge. You can override this setting using the HTML ALIGN property.
7. Check your document for errors, make any necessary changes, then save **Tent sidebar.htm** as a text document
8. Open **Tent sidebar.htm** in your Web browser

Trouble?

If you are using Netscape Navigator 4 and the sidebar is slow to appear, simply scroll down the page and back to the top to see the scrollbar more quickly.

As Figure L-14 shows, the text displays in a rectangle with a colored background to the right of the main text flow. Lydia specified absolute positioning settings for the graphics to position them just below the sidebar text.

FIGURE L-13: Web document containing style specifications for sidebar

```

float property creates sidebar effect
Code added to adjust layout for new text
Absolute position code for pack and bag icons

.tentbody {position: relative; left: 100px; width: 300px}
#backtext {position: absolute; left: 250px; font-size: 64pt; font-family: arial; color: #7093DB; z-index: -1}
#list {height: 300px; width: 600px; overflow: auto}
#sidebar {width: 350px; float: right; position: absolute; left: 400px; font-family: arial; font-size: 11pt; background: #8FBC8F}
.expl {width: 375px}
#pack1 {position: absolute; left: 505px}
#pack2 {position: absolute; left: 535px}
#pack3 {position: absolute; left: 425px}
#pack4 {position: absolute; left: 460px}
#bag1 {position: absolute; left: 565px}
#bag2 {position: absolute; left: 670px}
//-->
</STYLE>

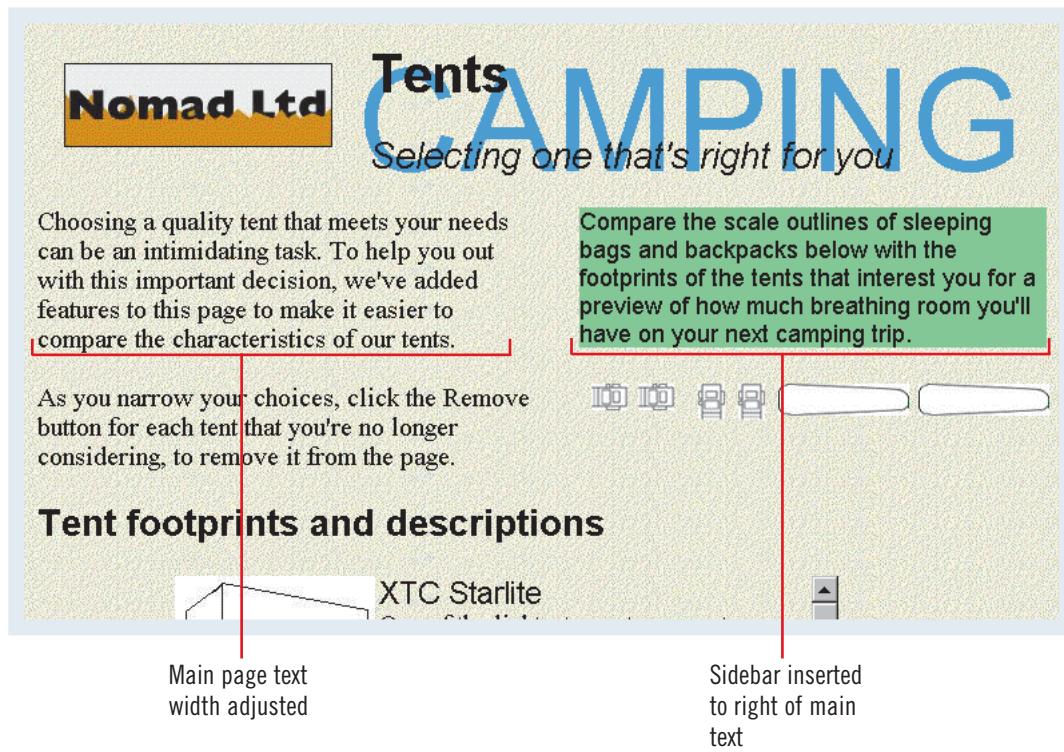
<SCRIPT LANGUAGE="javascript">
<!--
Nav4 = (document.layers) ? 1:0;
IE4 = (document.all) ? 1:0;

if(!IE4) {window.onerror=null}

totalTents = 0;
function countHeaders() {
    for (var i = 0; i < document.all.length; i++){


```

FIGURE L-14: Web page displaying sidebar





Incorporating an Advanced Positioning Function

By creating scripts to interact with position and layer information, you can add many advanced features to your Web pages. One exciting result of scripting position in Internet Explorer is dragging. A script enabling the drag feature can adjust the position of the selected element based on the coordinates of the pointer and then assign the element to its final position once the user releases the mouse button. This drag feature allows users to rearrange elements into an order that is more useful for them than the page's default organization or to interact with Web page models and games.  Lydia wants to let Internet Explorer users drag the scale outlines of sleeping bags and backpacks over the tent outlines so they can explore how much each tent holds.

Steps 123

1. Open the file **HTML L-7.htm** in your text editor, then save it as a text document with the filename **Tent drag.htm**

Notice that Lydia has created the class description .drag in the embedded style section for a draggable element.

2. Scroll down until the var elDrag=null code appears in the document window

Notice that Lydia has created a script in the header section that handles dragging of absolute-positioned elements.

3. Scroll down until the IMG SRC code for the first draggable image appears in the document window, select the text **[replace with drag code]** in the opening DIV tag for the image tag, then press **[Delete]**

4. Type **CLASS="drag" canDrag**

This code associates the image with the “drag” class that Lydia defined in the embedded style sheet, which specifies a z-index of 10. The script Lydia inserted earlier in the code identifies draggable images through the “canDrag” attribute.

5. Repeat Steps 2 and 3 for the remaining five IMG tags

Figure L-15 shows the draggable image codes for this page.

6. Check your document for errors, make any necessary changes, then save **Tent drag.htm** as a text document

7. Open **Tent drag.htm** in your Web browser

8. If you are using IE4, drag some of the backpack and sleeping bag outlines onto a tent footprint

Look at Figure L-16, and notice that the images move with the mouse pointer. The user can display the images on top of the tent outlines.

QuickTip

You also can format text elements to be draggable, using DIV or SPAN tags as containers.

FIGURE L-15: Web document containing draggable image codes

Image codes formatted for dragging

```
features to this page to make it easier to compare the characteristics of  
our tents.<BR><BR></DIV>  
  
<DIV ID="pack1" CLASS="drag" canDrag>  
<IMG SRC="pack.jpg">  
</DIV>  
  
<DIV ID="pack2" CLASS="drag" canDrag>  
<IMG SRC="pack.jpg">  
</DIV>  
  
<DIV ID="pack3" CLASS="drag" canDrag>  
<IMG SRC="pack2.jpg">  
</DIV>  
  
<DIV ID="pack4" CLASS="drag" canDrag>  
<IMG SRC="pack2.jpg">  
</DIV>  
  
<DIV ID="bag1" CLASS="drag" canDrag>  
<IMG SRC="bag.jpg">  
</DIV>  
  
<DIV ID="bag2" CLASS="drag" canDrag>  
<IMG SRC="bag.jpg">  
</DIV>
```

Attribute marks element as draggable

Class property associates z-index value with each element

FIGURE L-16: Web page showing dragged images

can be an intimidating task. To help you out with this important decision, we've added features to this page to make it easier to compare the characteristics of our tents.

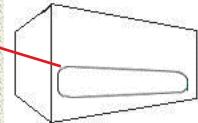
Bags and backpacks below will the footprints of the tents that interest you for a preview of how much breathing room you'll have on your next camping trip.

As you narrow your choices, click the Remove button for each tent that you're no longer considering, to remove it from the page.



Tent footprints and descriptions

Sleeping bag outline positioned on tent footprint



XTC Starlite

One of the lightest, most compact three-season tents available. Featuring two-pole clip design with a built-in vestibule.

[Remove Starlite](#)



Pack outline being dragged

Amano Brevifolia

Practice

► Concepts Review

Label the elements marked in figure L-17 with the CSS-P properties used to create them.



Match each term with its description.

- | | |
|-------------------------|---|
| 6. Absolute positioning | a. Properties for specifying element dimensions |
| 7. Relative positioning | b. Places element at fixed coordinates outside page flow |
| 8. Float | c. Properties for specifying location |
| 9. Height and width | d. Property used to create sidebar |
| 10. Top and left | e. Places element relative to parent element's coordinates within page flow |

Select the best answer from the list of choices.

11. An absolutely positioned element is located

- a. On a separate layer from the rest of the Web page, offset from the parent element.
- b. On a separate layer from the rest of the Web page, offset from the top-left corner of the browser window.
- c. On the same layer as the rest of the page, offset from the parent element.
- d. On the same layer as the rest of the page, offset from the top-left corner of the browser window.

12. DHTML allows precise positioning of page elements through an extension of Cascading Style Sheets called

- a. Absolute positioning.
- b. Cascading Style Sheets - Positioning.
- c. The "Position" style property.
- d. Tables.

13. The browser always places absolutely positioned text

- a. At the top-left corner of the browser window.
- b. At the specified coordinates relative to the top-left corner of the browser window.
- c. Behind the main page elements in z-index.
- d. At the specified coordinates relative to its parent element.

14. Which is not a valid measurement unit for specifying dimension and coordinate properties?

- a. Feet
- b. Inches
- c. Points
- d. Pixels

15. As you add new layers to a Web page, the elements on the most recent layers receive

- a. The same z-index value as earlier layers.
- b. Smaller z-index values than earlier layers.
- c. Larger z-index values than earlier layers.
- d. Negative z-index values.

16. Which property allows you to add scroll bars to specific elements?

- a. Scroll bar
- b. Float
- c. Layer
- d. Overflow

17. When adding a scroll bar to an element, assigning the value "auto" results in

- a. A scroll bar only when the element size requires it.
- b. Both horizontal and vertical scroll bars always appearing .
- c. No scroll bars.
- d. Addition of a vertical scroll bar only.

► Skills Review

1. Position an element absolutely.

- a. Open the file HTML L-8.htm in your Web browser.
- b. Open the file HTML L-8.htm in your text editor, then save it as a text document with the filename Pack absolute position.htm.
- c. Scroll to the top of the document's HEAD section, select the text [replace with logo absolute position code] in the embedded style sheet, then press [Delete].
- d. Type #logo {position: absolute; top: 30px; left: 30px}
- e. Scroll and select the text [replace with ID] in the opening DIV tag above the IMG tag for nomad.jpg, then press [Delete].
- f. Type ID="logo"
- g. Check your document for errors, make any necessary changes, then save Pack absolute position.htm as a text document.
- h. Open Pack absolute position.htm in your Web browser.

2. Position an element relatively.

- a. Open the file HTML L-9.htm in your text editor and save a copy as Pack relative position.htm.
- b. In the embedded style sheet, select the text [replace with head relative position code], then press [Delete].
- c. Type #head {position: relative; left: 275px}
- d. Scroll and select the text [replace with ID], then press [Delete].
- e. Select the adjacent text ALIGN="center" and press [Delete].
- f. Type ID="head"
- g. Check your document for errors, make any necessary changes, then save Pack relative position.htm as a text document.
- h. Open Pack relative position.htm in your Web browser.

3. Size an element manually.

- a. Open the file HTML L-10.htm in your text editor, and save it as a text document with the filename Pack element size.htm.
- b. Select the text [replace with packbody class description] in the embedded style sheet, then press [Delete].
- c. Type .packbody {position: relative; left: 100px; width: 300px}
- d. Check your document for errors, make any necessary changes, then save Pack element size.htm as a text document.
- e. Open Pack element size.htm in your Web browser and scroll down to the pack descriptions.

4. Stack screen elements.

- a. Open the file HTML L-11.htm in your text editor, then save it as a text document with the filename Pack layers.htm.
- b. In the embedded style sheet, select the text [replace with backtext layer code], then press [Delete].
- c. Type #backtext {position: absolute; left: 275px; font-size: 64pt; font-family: arial; color: #7093DB; z-index: -1}
- d. Scroll and select the text [replace with background text], then press [Delete].
- e. Type <DIV ID="backtext"> and press [Enter].
- f. Type HIKING and press [Enter], then type </DIV>
- g. Check your document for errors, make any necessary changes, then save Pack layers.htm as a text document.
- h. Open Pack layers.htm in your Web browser.

5. Add a scroll bar.

- a. Open the file HTML L-12.htm in your text editor, and save it as a text document with the filename Pack scroll.htm.
- b. In the embedded style sheet, select the text [replace with list scroll code], then press [Delete].
- c. Type `#list {height: 300px; width: 600px; overflow: auto}`
- d. Scroll and select the text [replace with opening DIV tag], then press [Delete].
- e. Type `<DIV ID="list">`
- f. Scroll, and select the text [replace with closing DIV tag], press [Delete], then type `</DIV>`
- g. Check your document for errors, make any necessary changes, then save Pack scroll.htm as a text document.
- h. Open Pack scroll.htm in your Web browser, and scroll down to view the list of pack outlines and descriptions.
- i. Use the scroll bar for the list to view all of the pack descriptions.

6. Create a sidebar.

- a. Open the file HTML L-13.htm in your text editor, then save it as a text document with the filename Pack sidebar.htm.
- b. In the embedded style sheet, select the text [replace with sidebar code], then press [Delete].
- c. Type `#sidebar {width: 350; float: right; position: absolute; left: 400px; font-family: arial; font-size: 11pt; background: #8FBC8F}`
- d. Scroll and select the text [replace with ID], then press [Delete].
- e. Type `ALIGN="left" ID="sidebar"`
- f. Check your document for errors, make any necessary changes, then save Pack sidebar.htm as a text document.
- g. Open Pack sidebar.htm in your Web browser.

7. Incorporate an advanced positioning function.

- a. Open the file HTML L-14.htm in your text editor, then save a copy as Pack drag.htm.
- b. Scroll and select the text [replace with drag code] in the first image tag, then press [Delete].
- c. Type `CLASS="drag" canDrag`
- d. Repeat Steps b and c for the remaining five IMG tags.
- e. Check your document for errors, make any necessary changes, then save Tent drag.htm as a text document.
- f. Open Tent drag.htm in your Web browser.
- g. If you are using Internet Explorer, drag some of the water bottle and tent outlines onto a backpack outline.

► Independent Challenges

- 1.** You are revising the Popular supplies page for the Green House plant store. You have incorporated positioning using style sheets to improve the appearance of the Web page and to make sure it displays similarly in different screen resolutions. Also, you have removed some of the features you added earlier to keep the page from overwhelming users with features. The owners would like to add additional information to the page about how to use it. You think this would fit best in a sidebar next to the headings.

To complete this independent challenge:

- Open the file HTML L-15.htm in your text editor, then save a copy as “Green House supplies with sidebar.htm”.
- In the embedded style sheet, select the text [replace with #sidebar definition], then press [Delete].
- Type `#sidebar {position: relative; width: 25%; float: right; font-family: arial; font-size: 11pt; background: #8FBC8F}`
- Scroll down to the opening DIV tag for the sidebar text, select the text [replace with ID], then press [Delete].
- Type `ALIGN="left" ID="sidebar"`
- Check your document for errors and make any necessary changes, save Green house supplies with sidebar.htm as a text document and close it, then open it in your Web browser. If necessary, edit your code until the page displays appropriately.

- 2.** You are adding CSS positioning information to the rider tips page for Sandhills Regional Public Transit. To superimpose part of the heading text on the logo graphic, you would like to rearrange the heading section.

To complete this independent challenge:

- Open the file HTML L-16.htm in your Web browser, then explore the page.
- Open the file HTML L-16.htm in your text editor, then save a copy as “SRPT positioned rider tips.htm”.
- Create an embedded style sheet, define a style in the embedded style sheet named `#logo` specifying absolute position, top at 25px, left at 25px, and z-index of -1, then scroll down to the opening DIV tag above the IMG tag for bus.jpg and replace the text [replace with ID] with `ID="logo"`.
- Define a style in the embedded style sheet named `#headtop` specifying absolute position, left at 30px, and top at 25px, then scroll down to the opening DIV tag for the first line of the page heading and replace the text [replace with ID] with `ID="headtop"`.
- Define a style in the embedded style sheet named `#headbtm` specifying absolute position, left at 60 px, and top at 325px, then scroll down to the opening DIV tag for the second line of the page heading and replace the text [replace with ID] with `ID="headbtm"`.
- Define a style in the embedded style sheet named `#sidetext` specifying absolute position, text floating on the right, left at 400px, top at 25px, and width of 350px, then scroll down to the opening DIV tag for the instructions section and replace the text [replace with ID] with `ALIGN="left" ID="sidetext"`.
- Add style code to the UL tag for the bulleted list specifying relative position and top at 0px.
- Check your document for errors and make any necessary changes, save and close it, then open it in your Web browser. If necessary, edit your code until the page displays appropriately.

- 3.** You are updating Web pages for the Community Public School Volunteers organization by adding CSS positioning information. You want to improve the layout by placing the CPSV logo graphic behind the heading text.

To complete this independent challenge:

- a. Explore the file HTML L-17.htm in your Web browser, then explore the file.
- b. Open HTML L-17 in your text editor, then save a copy as "CPSV positioned home.htm".
- c. Create a style in the embedded style sheet to place the new, widened logo (which is the colored letters CPSV) behind the heading text at the top of the page with absolute positioning. Use the left and top properties to position it so that it appears centered behind the text. (*Hint:* The heading is positioned relatively to allow it to be displayed in front of the logo in Navigator. Be sure to set a z-index for the logo to place it behind the text.) Reference the style you created in the opening DIV tag for the logo graphic, which is named cpsvlog3.jpg.
- d. Create a style in the embedded style sheet for the DIV tagset enclosing the H3 heading and the links to narrow the width of the text area to 50% of the page width, position the section relatively, 25% to the left, change the background color to white (#FFFFFF), and add a 1pt black solid border (the style code for this feature is border: 1pt black solid). Reference the style you created in the opening DIV before the <H3> heading.
- e. Save your changes, preview the page in your Web browser, and make any changes necessary to improve the appearance of the Web page.



- 4.** The World Wide Web Consortium (W3C) has included a new set of positioning specifications in its revised Cascading Style Sheet guidelines, known as CSS2. Many of the new features that the W3C has outlined are supported in Microsoft and Netscape fifth-generation browsers. To complete this independent challenge, log on to the Internet, open your Web browser, and use a search engine to locate and open the Web site for the W3C. Locate information about the CSS2 guidelines, then print details of two new positioning features in CSS2. Write a paragraph on each feature, including what it allows a Web programmer to do and suggested syntax for implementing it. Check the Microsoft and Netscape Web sites to find out if their fifth-generation browsers will support either of the features you selected and include this information in your paragraphs. Submit your printout and paragraphs to your instructor.

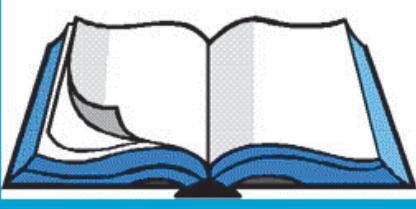
► Visual Workshop

You are improving the layout of the Web pages for Touchstone Booksellers using CSS positioning. Using the file HTML L-18.htm as a starting point, position the heading elements to create the layout in Figure L-18. (*Hint:* Think of the text blocks on the left and right sides of the graphic as two floating sidebars. You should add code only to the two sidebar text blocks.) Save your changes as a text document with the filename Touchstone positioned home.htm.

FIGURE L-18

Touchstone Booksellers

*Specializing
in nonfiction
of all types*



*a locally-owned,
independent
bookstore since
1948*

You can use our Web site to search our current stock, place an order, request a search for an out-of-print book, or find out about upcoming events at our store.

- [Search our stock](#)
- [Place an order](#)